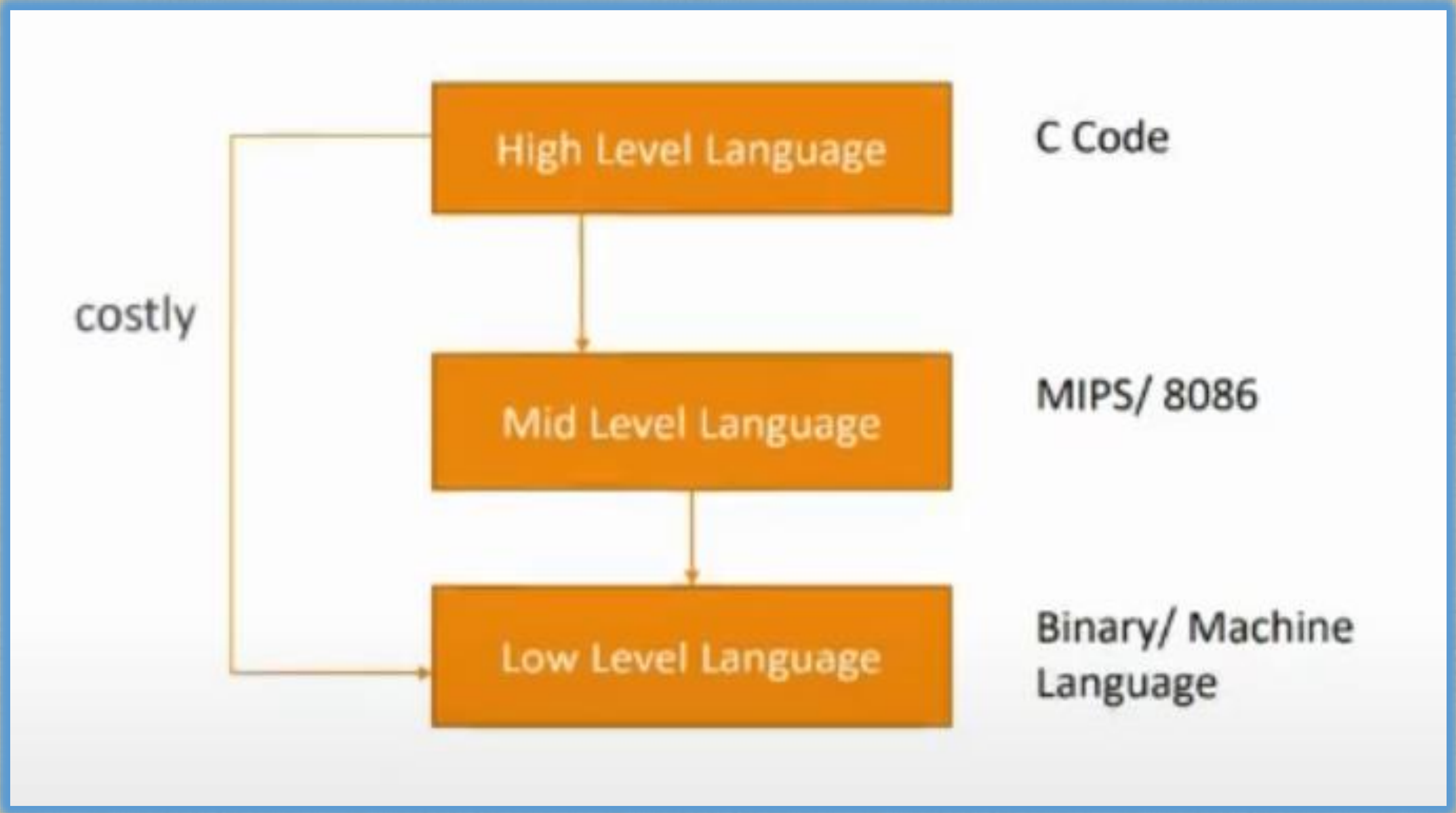
Computer Architecture

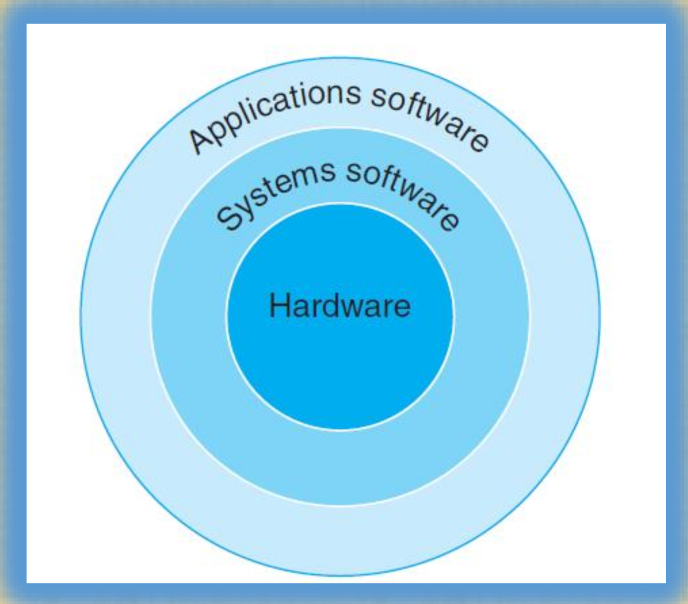
What We Learn?

- How Computers Work
  - MIPS instruction set architecture (ISA)
  - The implementation of MIPS instruction set architecture –MIPS processor design
- Issues Affecting Modern Processors
  - **Pipelining** –processor performance improvement
  - **Cache**–memory system, I/O systems


High Level Language
Mid Level Language
Low Level Language
costly
C Code
MIPS/ 8086
Binary/ Machine Language

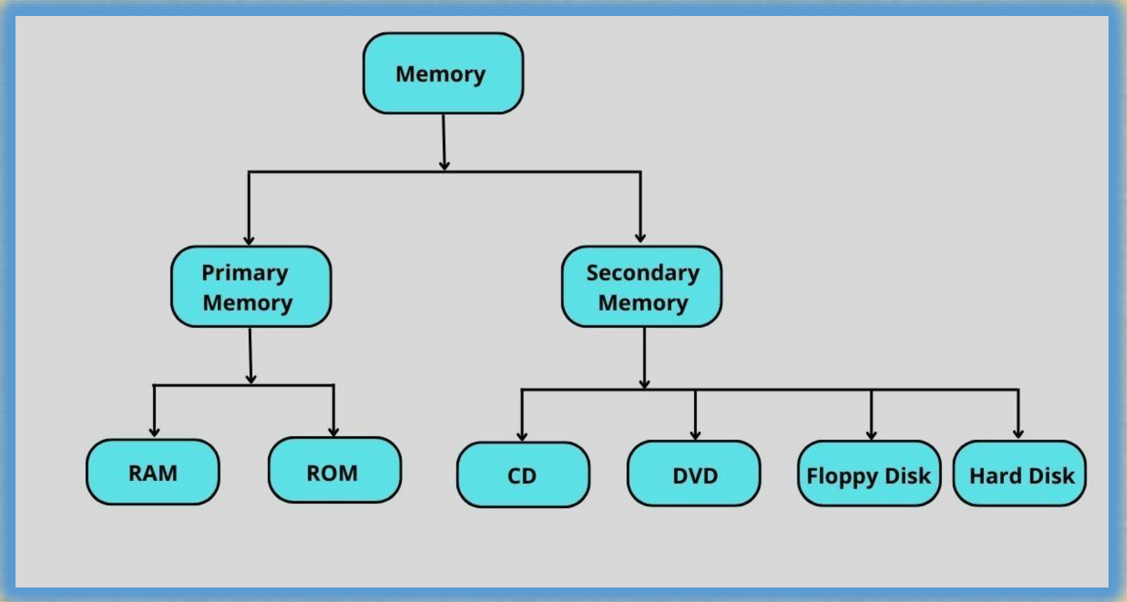
Software

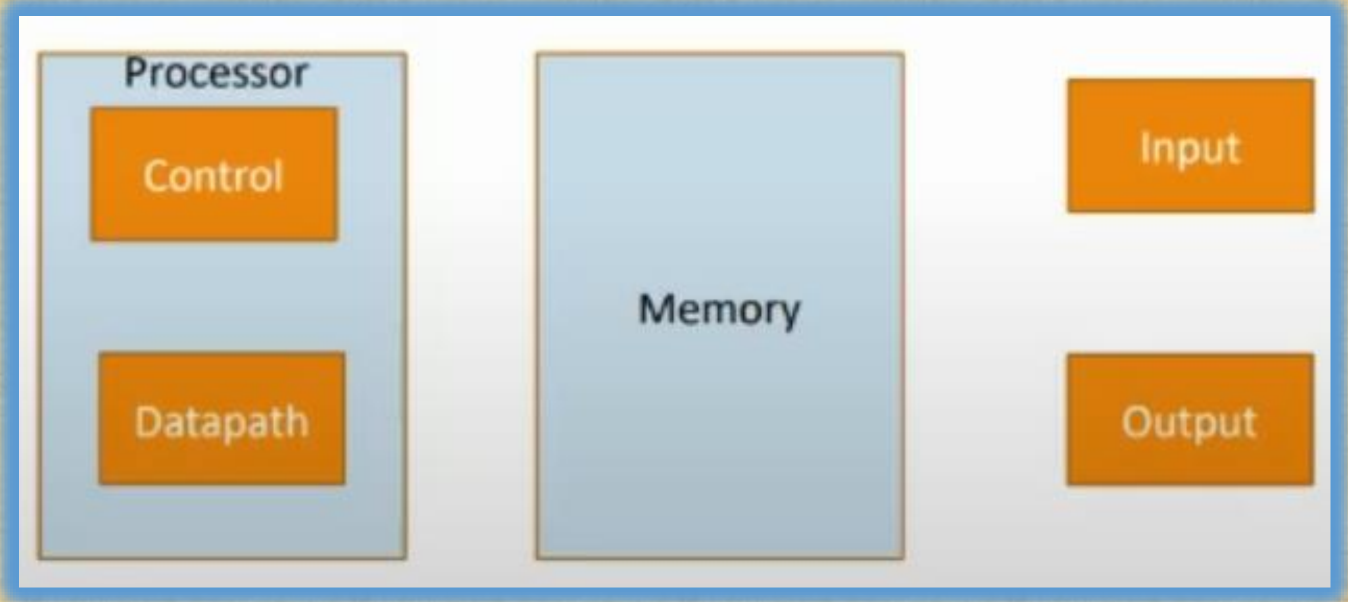
- Application software –
  - Word Processors, Email, Internet Browsers, Games
- Systems software –
  - Compilers, Operating Systems, device drivers



Hardware

- CPU (Central Processing Unit)
- Memory (RAM, ROM, Storage)
- I/O devices (mouse, keyboard, Monitor)





Operating System (OS)

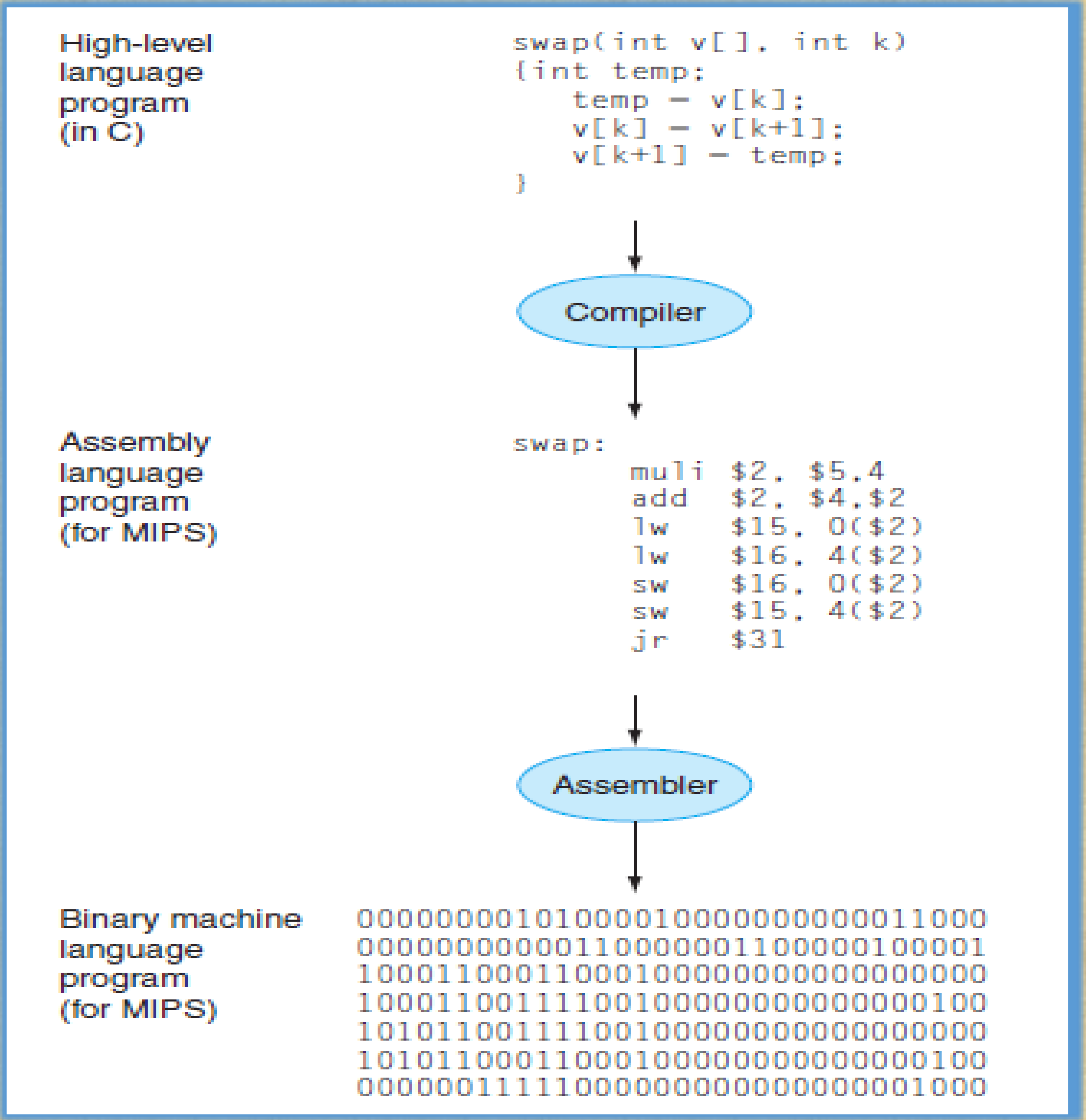
- **Definition**:
  - Interfaces between a user’s program and the hardware and provides a variety of services and supervisory functions.
- **Functions:**
  - handling basic input and output operations.
  - allocating storage and memory.
  - providing for sharing the computer among multiple applications using it simultaneously.

- Examples of operating systems: **Windows, Linux, and MacOS**

Compiler

- The translation of a program written in a high-level language, such as C or Java, into instructions that the hardware can execute.
- The translation from a high-level language program to hardware instructions is complex.

Microprocessor
without
Interlocked
Pipeline
Stages (**MIPS**)

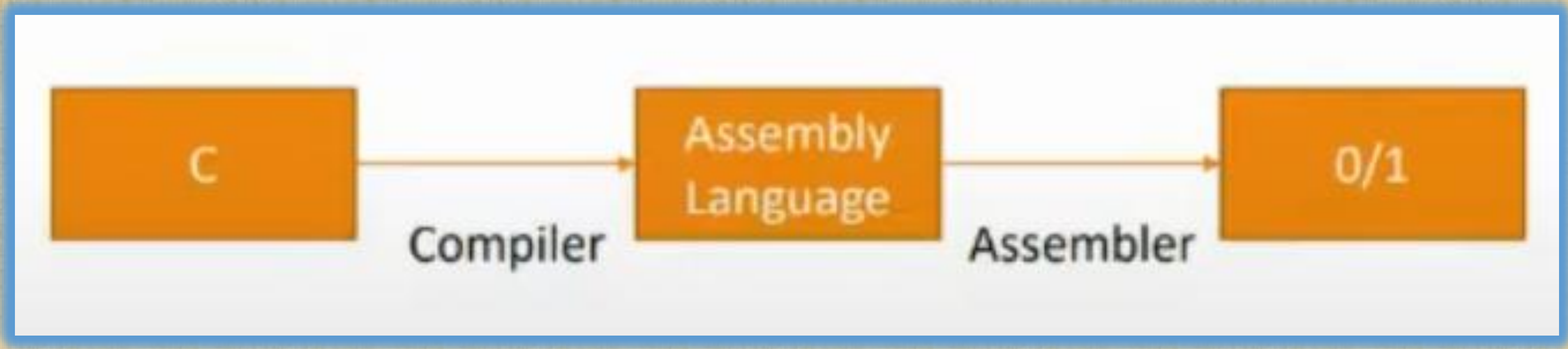

High-level language program (in C)
Compiler
Assembly language program (for MIPS)
Assembler
Binary machine language program (for MIPS)

```
swap(int v[], int k)
{int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

```
swap:
      muli $2, $5,4
      add  $2, $4,$2
      lw   $15, 0($2)
      lw   $16, 4($2)
      sw   $16, 0($2)
      sw   $15, 4($2)
      jr   $31
```

```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

- Assembler
  - A program that translates a symbolic version of instructions into the binary version.
- Assembly Language
  - A symbolic representation of machine instructions.



- *SUBSCRIBE:* Learn With Mahfuz
- *Follow our FB Page:* Learn With Mahfuz
- *Join our FB Community:* Learn With Mahfuz Community
- *Follow our LinkedIn Page:* Learn With Mahfuz
- *Follow me on GitHub:* mahfuzhasanreza